

# 日常生活のログ情報管理・活用に関する研究

北崎茂<sup>†</sup>

東京工業大学情報理工学研究科

望月祐洋<sup>‡</sup>

東京工業大学学術国際情報センター

## 1 はじめに

近年、インターネット利用ユーザ数とサービスプロバイダ数の増加により、ブログ (weblog) や SNS (Social Network Service) を始めとするユーザ参加型のコンテンツ制作 (CGM: Consumer Generated Media) が活発に行われている。コンテンツの種類は、ビジネスなどに関する実用的なものから個人の備忘録に至るまで実に多様である。しかし、現在の CGM で利用されているコンテンツ管理システム (CMS: Content Management System) はユーザの能動的な参加を要求するため、蓄積されるコンテンツは制作者が重要だと感じた情報に限られる。だが一般に、アイデアの創発や企業のマーケティングで重要となるのは、人々の日常的な行動や何気ない興味を把握することで得られる潜在的な情報である。こうした情報を収集するためにアンケート調査が行われるが、この場合も、記憶の曖昧さや主観的な判断、そして調査に要する手間の問題により、潜在的な情報を素早く的確に引き出せないことが多い。

そこで、日常生活の中でセンサや携帯機器を利用して、コンテンツ制作者の主観によらない潜在的な情報を収集するアプローチが考えられる。携帯電話の高機能化に顕著のように、各種のガジェットを安価に入手できるようになったためである。しかし、管理する機器が増加すれば、保存してあるコンテンツの管理だけでなく機器自体の管理も煩雑になる。

本研究では、潜在的な情報を収集および管理する上で、既に膨大なコンテンツを有する Web のアーキテクチャと親和性が高く、ユーザフレンドリであるシステムの構築を目的とする。本システムの特徴は、新しいコンポーネントを追加することで従来のシステムを変更することなく既存のコンテンツを利用し、ユーザが意識せずとも複数の機器により収集された日常的な行動の記録を管理できるようにすることである。本稿では、システムの概要とユーザ同士の通信手法の設計について報告する。

The Content Management and Utilization on Daily Work

<sup>†</sup>Shigeru Kitazaki (Tokyo Institute of Technology)

<sup>‡</sup>Masahiro Mochizuki (Tokyo Institute of Technology)

## 2 システム設計の指針

### 2.1 実装方法の選択

Web は急速な広がりを見せ、測定不能な数のリソースが存在する。リソースは単なる文書のような静的な情報資源だけに止まらず、CGM により人間をも巻き込むようになってきた。このような超巨大アーキテクチャを再構築することは現実的ではないし、別個に独自システムを構築することはユーザビリティに欠ける。ゆえに、Web をプラットフォームと考え、この上にシステムを実装する。この選択は、複数の機器でデータを連動させる場合に大きな利点になる。

Web 上では通信プロトコルとして HTTP (Hyper Text Transfer Protocol) が用いられている。HTTP は平易なプロトコルであるため、PC のみならず携帯電話やゲーム機からも Web 上のリソースにアクセスできる。しかし、HTTP はノードの状態を保存できないため、更新情報を取得するためには定期的にサーバにアクセスしなくてはならない。この方法の場合、サーバにアクセスした時に更新情報が必ずあるとは限らない、アクセスする時間間隔が一定であるために即座には更新を取得できないといった問題点がある。そこで、専用のアプリケーションとプロトコルを用いることが考えられるが、この場合は、利用する機器ごとにそれぞれの設定が必要となる。Web を利用する場合にも、専用のアプリケーションを利用する場合にも、それぞれ一長一短がある。

そこで、Comet[1] を利用して Web アプリケーションの欠点を解消することが考えられる。この方法の場合、現在 Web で幅広く使われている REST[2] アーキテクチャを利用できるため、従来のシステムを変更することなく既存のコンテンツを利用できる。

### 2.2 ログ情報

本システムで取り扱う「日常的な行動」として、1) 購買活動、2) マルチメディアコンテンツの視聴、3) 文書管理、を想定し、それぞれの行動履歴をログ情報として扱う。いずれもモジュール化されているため、別途ログ情報を管理したい行動があれば適宜追加できる。

ログ情報は PC や携帯機器，電子マネーなどにより収集し，ユーザ固有のデータベースに蓄積する．蓄積されたログ情報は自分で眺めるだけでなく，適切なアクセス権を設定することで公開もできる．

既存のマルチメディアコンテンツや文書も，リソースの位置を指定することでシステムへのコピーまたはメタデータによる関連付けを行うことができる．リソースは Web 上でもローカルファイルシステム上でも良い．

## 2.3 ログ情報の管理・活用

本システムでは，購買活動やマルチメディア視聴のログレポートを作成する．ブログが備忘録として利用されている現実を考えると，こうした自己を顧みる行為も必要と言える．ただし，本システムでは単一の情報ではなく，多様なログ情報を組み合わせる時系列に沿って表示する．

コンテンツには文書も含むため，アクセス権を変更することで配信もできる．文書の公開配信は既存の Web と変わらないが，Comet を利用することでリアルタイム性が加わる．予め設定したアクセス権に従って機械的に配信するだけでなく，データを要求してきたユーザに合わせて異なるコンテンツをアドホックに入手で変更できる．

一般に，購買活動に関するログ情報のように個人のプライバシーに関わるセンシティブな情報は公開しないが，一緒にいる仲間には見せたい部分もある．本システムでは，リアルタイムにアクセス権を変更することでこの問題を解消し，ログ情報を他者と比較することを可能とする．他人のログ情報を眺めることで，相手の意外な一面を発見したり，相手にとっては些細なことであったため表層化しなかったが，閲覧者にとっては有益な情報を発見することが期待される．

## 3 システム設計

本システムの概略を図 1 に示す．それぞれの通信は HTTP を使用し，データは XML で管理する．広く普及している技術を利用するため，既存のシステムにコンポーネントを追加することでサービスを実現できる．なお，Web サーバと DepotServer は XML-RPC を利用して通信するため，稼動するマシンは同一でも異なっても良い．

それぞれのコンポーネントは次の役割を果たす．DepotServer ユーザ (管理者) と 1 対 1 に対応し，常時起動しておく必要がある．ストレージには，ユーザのログ情報および所持あるいは関与しているコンテンツのメタデータを保存する．コンテンツの保存場所は，Web 上でもローカルファイルシステム上でも良い．管

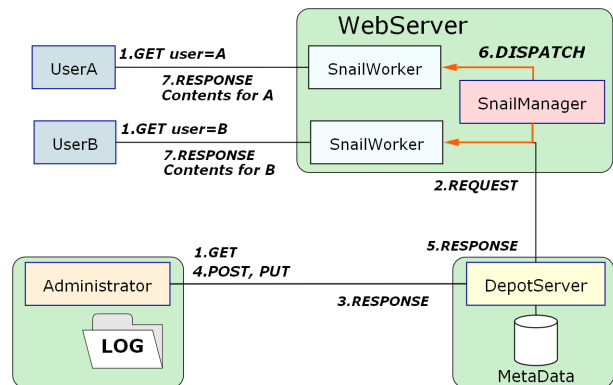


図 1: システム図

理者の意思によりローカルファイルシステム上のコンテンツ配信もできる．

SnailManager 複数の SnailWorker の状態を管理し，DepotServer の管理者が設定した応答規則に従って SnailWorker にコンテンツを送信させる．コンテンツは DepotServer から取得する．

SnailWorker クライアントからのリクエストを管理する．SnailManager からの許可がない場合はレスポンスを返さない．

## 4 システム実装

実装にはスクリプト言語 Python を用いた．DepotServer, SnailManager は通信処理を記述するためのフレームワークである Twisted[3] を利用した．SnailWorker は現行のウェブサーバへの追加を容易にするため，CGI として実装した．

## 5 まとめ

本稿では，ユーザ同士のインタラクションにより潜在的な情報の表層化を支援するシステムを提案した．本システムの特徴は，REST アーキテクチャを利用することで，現行システムの変更なく既存のコンテンツも新たなコンテンツも管理できることである．

今後は，ユーザに意識させずに複数の機器でコンテンツを同期させ，効率的に記録を管理する手法について検討を進めたい．

## 参考文献

- [1] Alex Russell: “Comet: Low Latency Data for the Browser,” <http://alex.dojotoolkit.org/?p=545>
- [2] Roy T. Fielding: “Architectural styles and the design of network-based software architectures,” PhD Thesis, University of California, Irvine, 2000.
- [3] Abe Fettig: “Twisted Network Programming Essentials,” O’Reilly, 2005.